

07-03-00

A

IN THE UNITED STATES
PATENT AND TRADEMARK OFFICE

PATENT APPLICATION

John P Hearn
Kim N Matthews
Christopher C Yu

CASE 1-21-1

TITLE MPEG Flow Identification For IP Networks

ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

SIR:

NEW APPLICATION UNDER 37 CFR § 1.53(b)

Enclosed are the following papers relating to the above-named application for patent:

Specification
2 Formal Sheets of drawing(s)

CLAIMS AS FILED				
	NO. FILED	NO. EXTRA	RATE	CALCULATIONS
Total Claims	43 - 20 =	23	x \$18 =	\$414
Independent Claims	7 - 3 =	4	x \$78 =	\$312
Multiple Dependent Claims, if applicable			+ \$260 =	\$0
Basic Fee				\$690
TOTAL FEE				\$1416

Please file the application and charge **Lucent Technologies Deposit Account No. 12-2325** the amount of \$1416, to cover the filing fee. Duplicate copies of this letter are enclosed. In the event of non-payment or improper payment of a required fee, the Commissioner is authorized to charge or to credit **Deposit Account No. 12-2325** as required to correct the error.

The Assistant Commissioner for Patents is hereby authorized to treat any concurrent or future reply, requiring a petition for extension of time under 37 CFR § 1.136 for its timely submission, as incorporating a petition for extension of time for the appropriate length of time if not submitted with the reply.

Please address all correspondence to **Docket Administrator (Room 3C-512), Lucent Technologies Inc., 600 Mountain Avenue, P.O. Box 636, Murray Hill, New Jersey 07974-0636**. However, telephone calls should be made to me at 908-582-4323.

Respectfully,

Eugene J Rosenthal

Eugene J Rosenthal
Reg. No. 36658
Attorney for Applicant(s)

Date: 6/30/00
Lucent Technologies Inc.
600 Mountain Avenue (Room 3C-512)
P.O. Box 636
Murray Hill, New Jersey 07974-0636

"Express Mail" mailing label
number: EK552897825US
Date of Deposit 6/30/00

I hereby certify that this Application is
being deposited with the United States Postal
Service "Express Mail Post Office to
Addressee" service under 37 CFR 1.10 on the
date indicated above and is addressed to the
Commissioner of Patents and Trademarks
Washington, D.C 20231

IRIS E. Thomas
(Printed name of person mailing paper or fee)
IRIS E. Thomas
(Signature of person mailing paper or fee)

MPEG FLOW IDENTIFICATION FOR IP NETWORKS

Technical Field

This invention relates to the art of transmitting real-time-constrained information over internet protocol (IP) networks, and more particularly, to identifying particular streams of real-time-constrained information, such as video encoded using Motion Pictures Expert Group (MPEG)-2 encoding, so that they may be given appropriate processing treatment.

Background of the Invention

A problem in the art of transmitting packets containing real-time-constrained information over internet protocol (IP) networks is the need to give each type of information the appropriate processing. In order to do this, it is necessary to know the type of information that is contained within each packet as it passes through the network at each point at which the packet will be processed. In particular, video, such as using Motion Pictures Expert Group (MPEG)-2 encoded video, cannot withstand dropped packets. Therefore, in processing streams of packets that contain MPEG-2 video, it is necessary to give priority to any MPEG-2 video streams over other streams which are less sensitive, or insensitive, to dropped packets.

Prior art techniques prioritize packets based on predefined criteria without actually ascertaining, e.g., by investigating the contents of the packet, that the packets actually contain MPEG-2 video. For example, a packet flow may be defined and it is assumed that all packets in that flow are MPEG-2 packets, and packets from that flow are treated as if they contain MPEG-2 packets without regard for their actual contents. A flow is often defined by specifying source and destination IP addresses for the flow, or by specifying the source/destination port address thereof.

Another prior art technique that is used to prioritize packets is based upon the so-called "type-of-service" (TOS) byte, which is part of the IP packet header. The TOS may be used to indicate a coarse prioritization, so that, for example, it is assumed that any packet with either a predefined value in the TOS byte, or a value in the TOS byte that is at least a predefined value, contains MPEG-2 video and is treated appropriately.

Because these prior art techniques do not actually investigate the contents of the packet to be certain that it contains MPEG-2 video, it is possible that packets that do not contain MPEG-2 video will be processed as if they contained MPEG-2 video. Provided that there is sufficient bandwidth in the processing system, processing these non-MPEG-2 packets, i.e., these fake MPEG-2 packets, as if they were indeed actual MPEG-2 packets

is not a problem. However, in the face of limited bandwidth—and what system does not have limited bandwidth—processing these fake MPEG-2 packets as undroppable actual MPEG-2 packets unnecessarily consumes system resources. Furthermore, these prior art prioritization arrangements can be abused by an unscrupulous user who sets his flow, or TOS byte, to indicate that he is sending MPEG-2 video, when in reality he is not.

In addition, setting up the flows in an IP network—the flows being a static configuration that specifies fixed ports—requires administration and/or reconfiguration each time a flow needs to be changed, i.e., a) when a point to point connection changes one of its points, b) when a new connection is made, or c) the like. Note that each switch or processing unit through which an MPEG-2 video stream flows must be updated with the new flow identifying information each time a flow needs to be changed. Thus, there is a constant administrative burden due to the resulting flow churn.

Another problem that arises in prior art arrangements using the TOS byte to define a flow that is to contain MPEG-2 video is the need for all of switches or processing units through which the flow passes to agree to a common TOS byte value, or set of values, that indicate MPEG-2 video. Otherwise, some switches or processing units may not properly handle, e.g., may drop, the MPEG-2 packets. It is difficult in practice to arrange for such agreement, especially when the flow travels over multiple networks.

Summary of the Invention

We have recognized that that the problems with the prior art can be overcome, in accordance with the principles of the invention, by actually determining that a packet contains MPEG-2 video rather than using predefined streams or priority levels that are assumed to contain such information as is done in the prior art. More specifically, in accordance with an aspect of the invention, the “sync” bytes of the MPEG-2 stream are searched for within the IP packet payload, and when a pattern indicative of the sync bytes is found the sync bytes are identified and the packet is determined to contain MPEG-2 video. The sync byte was defined in MPEG-2 for over-the-air broadcasting in order that a television receiver be able to synchronize the MPEG-2 transport stream packets. Note that the MPEG-2 video, e.g., the MPEG-2 transport stream packets, may or may not be incorporated within real time protocol (RTP) packets before being incorporated into the IP packets.

Brief Description of the Drawing

In the drawing:

FIG. 1 shows internet protocol (IP) packet that is of the general type for which a determination may be made as to whether or not it contains MPEG-2 video based only on the packet's IP data payload in accordance with the principles of the invention;

FIG. 2 shows an expanded version of a portion of the packet shown in FIG. 1; and

FIG. 3 shows an exemplary process, in flowchart form, for processing an IP packet to determine if it contains MPEG-2 video, in accordance with the principles of the invention.

Detailed Description

The following merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise various arrangements which, although not explicitly described or shown herein, embody the principles of the invention and are included within its spirit and scope. Furthermore, all examples and conditional language recited herein are principally intended expressly to be only for pedagogical purposes to aid the reader in understanding the principles of the invention and the concepts contributed by the inventor(s) to furthering the art, and are to be construed as being without limitation to such specifically recited examples and conditions. Moreover, all statements herein reciting principles, aspects, and embodiments of the invention, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof. Additionally, it is intended that such equivalents include both currently known equivalents as well as equivalents developed in the future, i.e., any elements developed that perform the same function, regardless of structure.

Thus, for example, it will be appreciated by those skilled in the art that the block diagrams herein represent conceptual views of illustrative circuitry embodying the principles of the invention. Similarly, it will be appreciated that any flow charts, flow diagrams, state transition diagrams, pseudocode, and the like represent various processes which may be substantially represented in computer readable medium and so executed by a computer or processor, whether or not such computer or processor is explicitly shown.

The functions of the various elements shown in the FIGs., including functional blocks labeled as "processors", may be provided through the use of dedicated hardware as well as hardware capable of executing software in association with appropriate software. When provided by a processor, the functions may be provided by a single dedicated processor, by a single shared processor, or by a plurality of individual processors, some of which may be shared. Moreover, explicit use of the term "processor" or "controller"

should not be construed to refer exclusively to hardware capable of executing software, and may implicitly include, without limitation, digital signal processor (DSP) hardware, read-only memory (ROM) for storing software, random access memory (RAM), and non-volatile storage. Other hardware, conventional and/or custom, may also be included.

5 Similarly, any switches shown in the FIGS. are conceptual only. Their function may be carried out through the operation of program logic, through dedicated logic, through the interaction of program control and dedicated logic, or even manually, the particular technique being selectable by the implementor as more specifically understood from the context.

10 In the claims hereof any element expressed as a means for performing a specified function is intended to encompass any way of performing that function including, for example, a) a combination of circuit elements which performs that function or b) software in any form, including, therefore, firmware, microcode or the like, combined with appropriate circuitry for executing that software to perform the function. The invention
15 as defined by such claims resides in the fact that the functionalities provided by the various recited means are combined and brought together in the manner which the claims call for. Applicant thus regards any means which can provide those functionalities as equivalent as those shown herein.

Unless otherwise explicitly specified herein, the drawings are not drawn to scale.

20 FIG. 1 shows internet protocol (IP) packet 101 that is of the general type for which a determination may be made as to whether or not it contains MPEG-2 video based only on the packet's IP data payload in accordance with the principles of the invention. More specifically, in accordance with an aspect of the invention, a search is performed for the "sync" bytes of the MPEG-2 stream within the IP packet data payload of packet
25 101. When a pattern indicative of the sync bytes is found, the packet is determined to contain MPEG-2 video. Otherwise, the packet is determined to contain information that is not MPEG-2 video.

Packet 101 has a series of headers which precede IP data payload 111. In particular, packet 101 contains IP header 105, which is 20 bytes long and unreliable
30 datagram protocol/transmission control protocol (UDP/TCP) header 107, which is 8 bytes long. Note that conventionally IP header 105 and UDP/TCP header 107 are conventionally grouped together and referred to as the header for the IP packet. They are shown independently in FIG. 1 for clarity of exposition and pedagogical purposes only.

Note that IP packet 101, as shown, is a UDP packet. This is because, as of this
35 writing, UDP is typically used for real-time streaming, as TCP requires end-to-end communication. Thus, the invention is described herein in terms of UDP packets in IP.

However, those of ordinary skill in the art will readily recognize how to apply the principles of the invention to TCP packets.

IP data payload 111 follows UDP/TCP header 109. The number of bytes that are contained within IP data payload 111 is flexible, ranging from 0 and up, although certain transmission arrangements, such as ethernet, may impose other limits. Optional real time protocol (RTP) header 109, if present, is 12 bytes long, and is within IP data payload 111.

FIG. 2 shows an expanded version of IP data payload 111, which, as packet 101 is a UDP packet, is a UDP data payload. FIG. 2 shows an example in which UDP data payload 111 is carrying MPEG-2 video. As indicated, RTP header 109 may precede the MPEG-2 video within UDP data payload 111.

Consistent with any size limitations placed on it, UDP data payload 111 may carry any arbitrary number of MPEG-2 transport stream packets 201. Each of MPEG-2 transport stream packets 201 is 188 bytes in length. By virtue of the definition of the MPEG-2 transport layer, the first byte of each of MPEG-2 transport stream packets 201 is always a so-called "sync" byte 203, which has the value of 0x47.

In accordance with the principles of the invention, due to the regular spacing of the sync byte, it is possible to search through UDP data payload 111 for the presence of the expected pattern of MPEG-2 video, i.e., a pattern of having a sync byte as the first byte of UDP data payload 111—after any optional RTP header—and thereafter having a sync byte at each byte position in UDP data payload 111 that is a multiple of 188. Although finding a sync byte as the first byte of UDP data payload 111, after any optional RTP header, gives a strong indication that the IP packet contains MPEG-2 video, and it is assumed that for UDP data payloads of length 188 for which the first byte has the value of a sync byte that the packet contains MPEG-2 video, preferably each potential sync byte position should be checked. This is because the confidence level that MPEG-2 video has actually been found increases substantially when each position indeed contains a sync byte value.

In the event that most of the expected positions contain a sync byte, but one or more do not, whether or not to declare the packet as one containing MPEG-2 video is at the discretion of the implementor when designing, or configuring, the system. For example, if only one position at which a sync byte would be expected was not a sync byte, and the IP packet was indicated to have a transmission error, then the implementor may decide to have the system still treat the packet as containing MPEG-2 video.

FIG. 3 shows an exemplary process, in flowchart form, for processing an IP packet to determine if it contains MPEG-2 video, in accordance with the principles of the invention. The process is entered in step 301 when an IP packet is received. Next, in

step 303, the packet is processed so there is a pointer that points to the UDP data payload within the packet, i.e., a pointer pointing within the packet is incremented to point to the UDP data payload. Thereafter, conditional branch point 305 tests to determine if the length of the UDP data payload is a multiple of 188. If the test result in step 305 is YES, indicating that there is no RTP header and that the length of the UDP data payload corresponds to a multiple of the length of MPEG-2 transport stream packet, control passes to conditional branch point 307, which tests to determine if the byte of the UDP data payload being pointed to by the pointer has the value of a sync byte, e.g., 0x47.

If the test result in step 307 is YES, control passes to step 309, which increments the pointer 188 bytes, i.e., the length of an MPEG-2 transport stream packet. This should result in the pointer pointing either at a) the beginning of the next MPEG-2 transport stream packet or to the end of the UDP data payload, which is also the end of the packet, provided the UDP data payload actually contains MPEG-2 video, or b) just a random byte when the UDP data payload does not actually contain MPEG-2 video. Thereafter, control passes to conditional branch point 311, which tests to determine if the end of the IP packet has been reached. If the test result in step 311 is NO, indicating that there yet remains additional portions of the UDP data payload to process, control passes back to step 307 to process the rest of the packet as described above. If the test result in step 311 is YES, control passes to step 313, and the IP packet is declared to be one containing MPEG-2 video, in accordance with an aspect of the invention. It may then be further processed accordingly. The process exits in step 315.

If the test result in step 307 is NO, indicating that either the first byte or another byte at a 188 multiple position does not have the value of a sync byte, control passes to step 315 and the process is exited.

If the test result in step 305 is NO, control passes to conditional branch point 317, which tests to determine if the length of the UDP data payload is equal to a multiple of 188 plus the length of the RTP header. If the test result in step 317 is YES, indicating that the UDP data payload may contain an RTP header and thereafter MPEG-2 transport stream packets, control passes to step conditional branch point 319, which tests to determine if the first bytes of the UDP data payload which correspond in length to an RTP header, have the characteristics of an RTP header, e.g., there is a video indicator in the location where the payload type field is expected. More specifically, the payload type field is 7 bits, with the definition for MPEG-2 transport stream data being 0x21.

If the test result in step 319 is NO, which indicates that there was no RTP header or that the header was not indicative of video, control passes to step 315 and the process is exited without declaring the IP packet to be one containing MPEG-2 video. If the test

result in step 319 is YES, indicating that an RTP header for video was found, control passes to step 321, in which the pointer is incremented to point to the first byte after the RTP header. Control then passes to conditional branch point 307 and the process continues as described above.

5 If the test result in step 317 is NO, indicating that the IP packet does not contain a whole number of MPEG-2 video transport stream packets, control passes to step 323, in which a counter variable COUNT, which is to be used as the offset into the UDP data payload, is set to 0. Next, conditional branch point 325 tests to determine if COUNT is equal to the sum of 188 and the length of an RTP header. If the test result in step 325 is
10 YES, indicating that all the positions that could potentially contain a sync byte of a first MPEG-2 video transport stream packet in the UDP data payload have been tested and not been found to be a sync byte, the process is exited in step 315, i.e., the packet is not declared to contain MPEG-2 video. If the test result in step 325 is NO, indicating that all the positions that could potentially contain a sync byte of a first MPEG-2 video transport stream packet in the UDP data payload have not been tested, control passes to conditional branch point 327 in which another pointer, PACKTPT, is set to the value of COUNT.
15

Thereafter, in step 329, the byte within the UDP data payload pointed to by PACKTPT is tested to determine if it has the value of a sync byte. If the test result in step 329 is NO, indicating that the byte currently pointed to by PACKTPT is not a sync
20 byte, control passes to step 331 in which COUNT is incremented, so that it will point to the next byte in the UDP data payload. Control then passes back to step 325, and the process continues as described above.

If the test result in step 329 is YES, indicating that the byte currently pointed to by PACKTPT indeed has the value of a sync byte, control passes to conditional branch point 333, which tests to determine if the remaining number of bytes in the packet from the place being pointed to by COUNT is less than 188. If the test result in step 333 is
25 YES, indicating that a whole MPEG-2 video transport stream packet cannot be contained within the UDP data payload, control passes to step 315 and the process is exited, i.e., the packet is not declared to contain MPEG-2 video. If the test result in step 333 is NO, indicating that a whole MPEG-2 video transport stream packet could be contained within the UDP data payload, control passes to step 335, in which PACKTPT is incremented by 188.
30

At this point, if the UDP data payload indeed contains MPEG-2 video and the byte currently pointed to by COUNT is a sync byte, the byte pointed to by PACKTPT
35 should also have the value of a sync byte, or at or beyond the end of the packet. To this end, conditional branch point 337 tests to determine if the end of the packet has been

5

[illegible]

What is claimed is:

1 1. A method for processing an internet protocol (IP) packet, comprising the step
2 of identifying that said packet contains motion picture expert group (MPEG)-2 video as a
3 function of only the contents of said IP data payload of said IP packet.

1 2. The invention as defined in claim 1 wherein said MPEG-2 video is in transport
2 stream format.

1 3. The invention as defined in claim 1 wherein said IP data payload contains at
2 least one real time protocol (RTP) packet which contains said MPEG-2 video.

1 4. The invention as defined in claim 1 wherein said IP data payload is a
2 unreliable datagram protocol (UDP) data payload.

1 5. The invention as defined in claim 1 wherein said IP data payload is a
2 transmission control protocol (TCP) data payload.

1 6. The invention as defined in claim 1 further comprising the step of processing
2 said IP packet with a priority assigned for packets containing video when said packet is
3 identified in said identifying step to contain video.

1 7. The invention as defined in claim 1 wherein said identifying step further
2 includes the steps of:
3 determining whether or not there exists within said IP data payload at least an
4 expected pattern of MPEG-2 sync bytes that is indicative of the presence of MPEG-2
5 video.

1 8. The invention as defined in claim 7 wherein said determining step further
2 comprises the step of:

3 comparing a first byte of said IP data payload after any real time protocol (RTP)
4 header to the value of an MPEG-2 sync byte; and

5 when the result of said comparing step is that said first byte of said IP data
6 payload has the same value as an MPEG-2 sync byte, declaring said IP packet to be an
7 MPEG-2 packet.

1 9. The invention as defined in claim 7 wherein said determining step further
2 comprises the step of:

3 comparing a first byte of said IP data payload after any real time protocol (RTP)
4 header to the value of an MPEG-2 sync byte; and

5 when the result of said comparing step is that said first byte of said IP data
6 payload is an MPEG-2 sync byte and the length of said IP data payload after any RTP
7 header is the same as the length of an MPEG-2 transport stream packet, declaring said IP
8 packet to be an MPEG-2 packet.

1 10. The invention as defined in claim 7 wherein said determining step further
2 comprises the step of:

3 terminating said process and indicating that said expected pattern does not exist in
4 said packet unless the length of said IP data payload or the length of said IP data payload
5 less the length of a real time protocol (RTP) header is an integral multiple of the length of
6 an MPEG-2 transport stream packet;

7 pointing a pointer to a byte in said IP data payload, said byte being a first byte of
8 said IP data payload when said length of said IP data payload is an integral multiple of
9 the length of an MPEG-2 transport stream packet and said byte being a first byte of said
10 IP data payload after the length of a real time protocol (RTP) header when said IP data
11 payload less the length of a real time protocol (RTP) header is an integral multiple of the
12 length of an MPEG-2 transport stream packet;

13 performing a comparison between said byte being pointed to with the value of an
14 MPEG-2 sync byte and declaring said IP packet as a candidate to be an MPEG-2 packet
15 when the result of a most recent comparison is that said pointed to byte of said IP data
16 payload has the same value as an MPEG-2 sync byte

17 adjusting said pointer to point to byte in said IP data payload that is offset toward
18 the end of said IP packet by the length of an MPEG-2 transport stream packet;

19 repeating said performing and adjusting steps so long as said most recently
20 executed performing step declared said IP packet as a candidate to be an MPEG-2 packet
21 and the end of said IP data payload is not yet reached; and

22 declaring said packet to be an MPEG-2 packet when the end of said IP data
23 payload is reached during an attempt to execute said adjusting step and said most recently
24 executed performing step declared said IP packet as a candidate to be an MPEG-2 packet.

1 11. The invention as defined in claim 7 wherein said expected pattern is an
2 MPEG-2 sync byte value spaced 188 byte positions apart.

1 12. The invention as defined in claim 7 wherein said expected pattern is an
2 MPEG-2 sync byte value spaced apart by the length of an MPEG-2 transport stream
3 packet.

1 13. The invention as defined in claim 7 wherein said determining step further
2 comprises the step of:

3 declaring said IP packet to be an MPEG-2 packet when a search over the length of
4 a real time protocol header and the length of one MPEG-2 transport stream packet finds
5 sync byte for which offset therefrom by the length of one MPEG-2 transport stream
6 packet there is another sync byte.

1 14. The invention as defined in claim 7 wherein said determining step further
2 comprises the step of:

3 declaring said IP packet to be an MPEG-2 packet when a search over the length of
4 a real time protocol header and the length of one MPEG-2 transport stream packet finds
5 the value of a sync byte for which offset therefrom at each integral multiple of the length
6 of one MPEG-2 transport stream packet there is the value of a sync byte until the end of
7 said IP packet is reached or exceeded.

1 15. The invention as defined in claim 7 wherein said determining step further
2 comprises the step of:

3 declaring said IP packet to be an MPEG-2 packet when a search over the length of
4 a real time protocol header and the length of one MPEG-2 transport stream packet finds
5 the value of a sync byte for which offset therefrom by the length of one MPEG-2
6 transport stream packet there is the end of packet.

1 16. The invention as defined in claim 7 wherein said each of said sync bytes has
2 a value of 0x47.

1 17. The invention as defined in claim 7 wherein said at least one expected
2 pattern is the value of an MPEG-2 sync byte as the first byte of said IP data payload.

1 18. The invention as defined in claim 7 wherein said expected pattern is the
2 value of an MPEG-2 sync byte as the first byte of said IP data payload and every 188
3 bytes thereafter till the end of said IP data payload.

19. The invention as defined in claim 7 wherein said at least one expected patterns includes at least one of the sets of patterns consisting of: a) the value of an MPEG-2 sync byte as the first byte of said IP data payload after the length of a real time protocol (RTP) header and said IP data payload after said RTP header length has a length of 188 bytes, b) the value of an MPEG-2 sync byte as the first byte of said IP data payload after the length of a real time protocol (RTP) header and every 188 bytes thereafter till the end of said IP data payload, c) the value of an MPEG-2 sync byte as the first byte of said IP data payload and every 188 bytes thereafter till the end of said IP data payload, d) the value of an MPEG-2 sync byte as the first byte of said IP data payload and said IP data payload has a length of 188 bytes.

20. A method for processing an internet protocol (IP) packet, comprising the steps of:
 searching through a payload of said IP packet for a pattern indicative of the presence of motion picture expert group (MPEG)-2 video; and
 indicating that said IP packet contains MPEG-2 video only if said pattern is found.

21. The invention as defined in claim 20 wherein said searching step further includes the step of:
 determining whether a payload of said IP packet has a length equal to an integral multiple of a length of an MPEG-2 transport stream packet either before or after subtracting from said payload length the length of an RTP head.

22. The invention as defined in claim 20 wherein said searching step further includes the steps of:
 determining that a payload of said IP packet has a length equal to a length of an MPEG-2 transport stream packet;
 comparing the value of a first byte at a first location within said packet with the value of an MPEG-2 sync byte; and
 signaling said pattern is found when the result of said comparing step is that said value of said first byte at said first location matches the value of said sync byte.

1 23. The invention as defined in claim 20 wherein said searching step further
2 includes the step of:

3 determining whether a payload of said IP packet has a length equal to an integral
4 multiple of a length of an MPEG-2 transport stream packet,

5 comparing the value of a first byte at a first location within said packet and each
6 byte at an offset of a length of an MPEG-2 transport stream packet therefrom with the
7 value of an MPEG-2 sync byte; and

8 signaling said pattern is found when the result of each comparison performed in
9 said comparing step is that said first byte of said packet being compared matches the
10 value of said sync byte.

1 24. The invention as defined in claim 20 wherein said searching step further
2 includes the step of:

3 determining whether a payload of said IP packet has a length equal to an integral
4 multiple of a length of an MPEG-2 transport stream packet,

5 comparing the value of a first byte at a first location within said packet and each
6 byte at an offset of a length of an MPEG-2 transport stream packet therefrom with the
7 value of an MPEG-2 sync byte; and

8 signaling said pattern is found when the result of a majority of comparisons
9 performed in said comparing step is that said first byte of said packet being compared
10 matches the value of said sync byte.

1 25. The invention as defined in claim 20 wherein said searching step further
2 includes the step of:

3 determining whether a payload of said IP packet has a length equal to an integral
4 multiple of a length of an MPEG-2 transport stream packet,

5 comparing the value of a first byte at a first location within said packet and each
6 byte at an offset of a length of an MPEG-2 transport stream packet therefrom with the
7 value of an MPEG-2 sync byte; and

8 signaling said pattern is found when the result of at least a majority of
9 comparisons performed in said comparing step is that said first byte of said packet being
10 compared matches the value of said sync byte and said packet was indicated to contain an
11 error.

1 26. The invention as defined in claim 20 wherein said payload is at least one of a
2 set of payloads within an IP packet, said set consisting of: a) an IP data payload, b) an
3 unreliable datagram protocol (UDP) data payload that does not include a real time
4 protocol (RTP) header, c) that portion of a UDP data payload after an RTP header that is
5 included in said UDP data payload, and d) a transmission control protocol (TCP) data
6 payload.

1 27. The invention as defined in claim 20 further comprising the step of
2 processing said IP packet with a priority assigned for packets containing video when said
3 indicating step indicates that said IP packet contains video.

1 28. The invention as defined in claim 20 wherein said pattern corresponds to the
2 value of an MPEG-2 sync byte regularly spaced from an initial position through the end
3 of said payload, said regular spacing being equal to the length of an MPEG-2 transport
4 stream packet, said initial position being located within the length of a real time protocol
5 header and the length of one MPEG-2 transport stream packet from a start of said
6 payload.

1 29. The invention as defined in claim 20 wherein said pattern corresponds to the
2 value of an MPEG-2 sync byte and the end of said IP packet spaced apart by the length of
3 one MPEG-2 transport stream packet, said initial position being located within the length
4 of a real time protocol header and the length of one MPEG-2 transport stream packet
5 from a start of said payload.

1 30. The invention as defined in claim 20 wherein said pattern corresponds to the
2 value of an MPEG-2 sync byte regularly spaced from an initial prescribed position, said
3 regular spacing being equal to the length of an MPEG-2 transport stream packet.

1 31. A method for processing an internet protocol (IP) packet, comprising the
2 steps of:

3 searching through a payload of said IP packet for a pattern indicative of the
4 presence of motion picture expert group (MPEG)-2 video; and

5 indicating that said IP packet contains MPEG-2 video when said pattern is most
6 likely found.

1 32. The invention as defined in claim 31 wherein said pattern corresponds to an
2 MPEG-2 sync byte regularly spaced from an initial prescribed position, said regular
3 spacing being equal to the length of an MPEG-2 transport stream packet.

1 33. The invention as defined in claim 32 wherein said initial prescribed position
2 is a position within the group of positions consisting of: a) the first byte in an IP data
3 payload of said packet, b) the first byte of a UDP payload of said IP packet, c) the first
4 byte after an RTP header of an unreliable datagram protocol (UDP) payload of said IP
5 packet, d) the first byte of a transport control protocol (TCP) payload of said IP packet,
6 and e) the first byte of a TCP payload after a header contained therein indicating real time
7 information is contained in said IP packet.

1 34. The invention as defined in claim 31 further comprising the step of
2 processing said IP packet with a priority assigned for packets containing video when said
3 indicating step indicates that said IP packet contains video.

1 35. A method for processing an internet protocol (IP) packet, comprising the steps
2 of:

3 searching through a payload of said IP packet for a pattern indicative of video;
4 and

5 indicating that said IP packet contains video when said pattern is found; and

6 indicating that said IP packet does not contain video when said pattern is not
7 found.

1 36. The invention as defined in claim 35 further comprising the step of
2 processing said IP packet with a priority assigned for packets containing video when it is
3 indicated that said IP packet contains video.

1 37. The invention as defined in claim 35 wherein said pattern corresponds to the
2 value of an MPEG-2 sync byte regularly spaced from an initial position through the end
3 of said payload, said regular spacing being equal to the length of an MPEG-2 transport
4 stream packet, said initial position being located within the length of a real time protocol
5 header and the length of one MPEG-2 transport stream packet from a start of said
6 payload.

1 38. The invention as defined in claim 35 wherein said pattern corresponds to the
2 value of an MPEG-2 sync byte regularly spaced from an initial prescribed position, said
3 regular spacing being equal to the length of an MPEG-2 transport stream packet.

1 39. The invention as defined in claim 38 wherein said initial prescribed position
2 is a position within the group of positions consisting of: a) the first byte in an IP data
3 payload of said packet, b) the first byte of a UDP payload of said IP packet, c) the first
4 byte after an RTP header of an unreliable datagram protocol (UDP) payload of said IP
5 packet, d) the first byte of a transport control protocol (TCP) payload of said IP packet,
6 and e) the first byte of a TCP payload after a header contained therein indicating real time
7 information is contained in said IP packet.

1 40. A computer program in the form of machine readable instructions, said
2 computer program being for causing a system including a processor to:
3 search through a payload of an internet protocol (IP) packet for a pattern
4 indicative of the video;
5 indicate that said IP packet contains video when said pattern is found; and
6 indicate that said IP packet does not contain video when said pattern is not found.

1 41. The invention as defined in claim 40 wherein said computer program further
2 causes said processor to
3 process said IP packet with a priority assigned for packets containing video when
4 said IP packet is indicated to contain video.

1 42. Apparatus comprising:
2 a processor;
3 a memory coupled to said processor for storing a received internet protocol (IP)
4 packet; and
5 a program store;
6 wherein said program store contains instructions for causing said processor to
7 search through a payload of an internet protocol (IP) packet for a pattern
8 indicative of the video;
9 indicate that said IP packet contains video when said pattern is found; and
10 indicate that said IP packet does not contain video when said pattern is not found.

1 43. Apparatus, comprising:
2 means for searching through a payload of an internet protocol (IP) packet for a
3 pattern indicative of video; and
4 means for indicating that said IP packet contains video when said pattern is found.

Abstract

The payloads of IP packets are checked to actually determine if a packet contains MPEG-2 video rather than using predefined streams or priority levels that are assumed to contain such information as is done in the prior art. More specifically, the “sync” bytes of the MPEG-2 stream are searched for within the IP packet payload, and when a pattern indicative of the sync bytes is found the sync bytes are identified and the packet is determined to contain MPEG-2 video. The sync byte was defined in MPEG-2 for over-the-air broadcasting in order that a television receiver be able to synchronize the MPEG-2 transport stream packets. Note that the MPEG-2 video, e.g., the MPEG-2 transport stream packets, may or may not be incorporated within real time protocol (RTP) packets before being incorporated into the IP packets.

FIG. 1

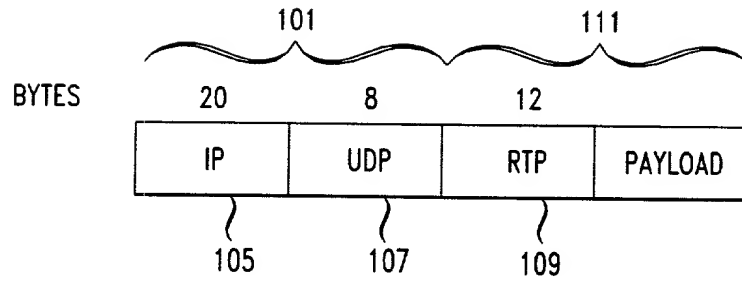
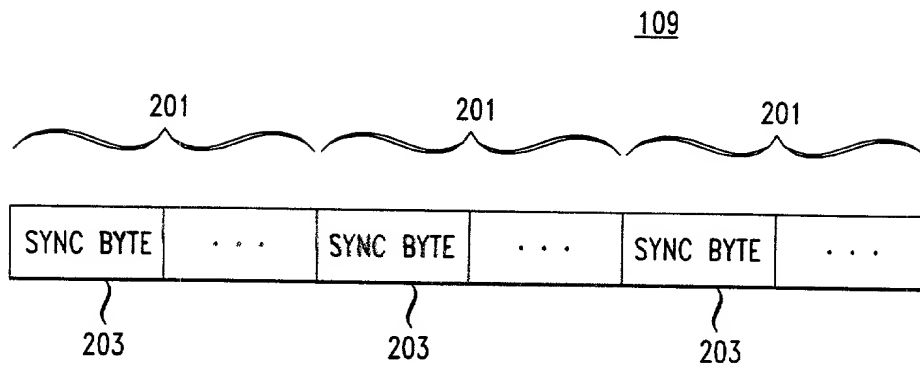


FIG. 2



1990-1991		1991-1992		1992-1993		1993-1994		1994-1995		1995-1996		1996-1997		1997-1998		1998-1999		1999-2000		2000-2001		2001-2002		2002-2003		2003-2004		2004-2005		2005-2006		2006-2007		2007-2008		2008-2009		2009-2010		2010-2011		2011-2012		2012-2013		2013-2014		2014-2015		2015-2016		2016-2017		2017-2018		2018-2019		2019-2020		2020-2021		2021-2022		2022-2023		2023-2024		2024-2025		2025-2026		2026-2027		2027-2028		2028-2029		2029-2030		2030-2031		2031-2032		2032-2033		2033-2034		2034-2035		2035-2036		2036-2037		2037-2038		2038-2039		2039-2040		2040-2041		2041-2042		2042-2043		2043-2044		2044-2045		2045-2046		2046-2047		2047-2048		2048-2049		2049-2050		2050-2051		2051-2052		2052-2053		2053-2054		2054-2055		2055-2056		2056-2057		2057-2058		2058-2059		2059-2060		2060-2061		2061-2062		2062-2063		2063-2064		2064-2065		2065-2066		2066-2067		2067-2068		2068-2069		2069-2070		2070-2071		2071-2072		2072-2073		2073-2074		2074-2075		2075-2076		2076-2077		2077-2078		2078-2079		2079-2080		2080-2081		2081-2082		2082-2083		2083-2084		2084-2085		2085-2086		2086-2087		2087-2088		2088-2089		2089-2090		2090-2091		2091-2092		2092-2093		2093-2094		2094-2095		2095-2096		2096-2097		2097-2098		2098-2099		2099-2100		2100-2101		2101-2102		2102-2103		2103-2104		2104-2105		2105-2106		2106-2107		2107-2108		2108-2109		2109-2110		2110-2111		2111-2112		2112-2113		2113-2114		2114-2115		2115-2116		2116-2117		2117-2118		2118-2119		2119-2120		2120-2121		2121-2122		2122-2123		2123-2124		2124-2125		2125-2126		2126-2127		2127-2128		2128-2129		2129-2130		2130-2131		2131-2132		2132-2133		2133-2134		2134-2135		2135-2136		2136-2137		2137-2138		2138-2139		2139-2140		2140-2141		2141-2142		2142-2143		2143-2144		2144-2145		2145-2146		2146-2147		2147-2148		2148-2149		2149-2150		2150-2151		2151-2152		2152-2153		2153-2154		2154-2155		2155-2156		2156-2157		2157-2158		2158-2159		2159-2160		2160-2161		2161-2162		2162-2163		2163-2164		2164-2165		2165-2166		2166-2167		2167-2168		2168-2169		2169-2170		2170-2171		2171-2172		2172-2173		2173-2174		2174-2175		2175-2176		2176-2177		2177-2178		2178-2179		2179-2180		2180-2181		2181-2182		2182-2183		2183-2184		2184-2185		2185-2186		2186-2187		2187-2188		2188-2189		2189-2190		2190-2191		2191-2192		2192-2193		2193-2194		2194-2195		2195-2196		2196-2197		2197-2198		2198-2199		2199-2200		2200-2201		2201-2202		2202-2203		2203-2204		2204-2205		2205-2206		2206-2207		2207-2208		2208-2209		2209-2210		2210-2211		2211-2212		2212-2213		2213-2214		2214-2215		2215-2216		2216-2217	
-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--

